A discontinuous Galerkin solver for front propagation

Olivier Bokanowski, Yingda Cheng, Chi-Wang Shu

A DISCONTINUOUS GALERKIN SOLVER FOR FRONT PROPAGATION

OLIVIER BOKANOWSKI, YINGDA CHENG, AND CHI-WANG SHU

Abstract. We propose a new discontinuous Galerkin (DG) method based on [9] to solve a class of Hamilton-Jacobi equations that arises from optimal control problems. These equations are connected to front propagation problems or minimal time problems with non isotropic dynamics. Several numerical experiments show the relevance of our method, in particular for front propagation.

1. Introduction

The Hamilton-Jacobi (HJ) equation

$$\varphi_t + H(\varphi_{x_1}, \ldots, \varphi_{x_d}, x_1, \ldots, x_d) = 0, \qquad \varphi(x,0) = \varphi^0(x) \tag{1}$$

arises in many applications, e.g., control theory and differential games. In this paper, we focus on the following Hamilton-Jacobi-Bellman (HJB) equation:

$$\varphi_t + \max_{\alpha \in \mathcal{A}} (f_\alpha(\mathbf{x}) \cdot \nabla \varphi) = 0, \quad t \geq 0, \ \ \mathbf{x} \in \mathbb{R}^d \tag{2a}$$

$$\varphi(0, \mathbf{x}) = \varphi^0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d \tag{2b}$$

where $\mathcal{A}$ is a non-empty compact set. It is equivalent to (1) by setting $H(p, \mathbf{x}) := \max_{\alpha \in \mathcal{A}}(f_\alpha(\mathbf{x}) \cdot p)$.

Equation (2) comes from the following optimal control problem:

$$\varphi(t, \mathbf{x}) := \inf\{\varphi^0(y_{\mathbf{x}}^\alpha(-t)), \ \alpha \in L^\infty((0,t), \mathcal{A})\}, \tag{3}$$

where $y = y_{\mathbf{x}}^\alpha$ is the absolutely continuous solution of $\dot{y}(\theta) = f_{\alpha(\theta)}(y(\theta))$ for almost every $\theta \in [-t; 0]$ and with $y(0) = \mathbf{x}$ (see for instance [3]).

The resolution of (2) is also motivated by a general type of non-isotropic front propagation problems. By front propagation we mean that we focus on the computation of the 0-level set $\{\mathbf{x}, \ \varphi(t, \mathbf{x}) = 0\}$. Then, Eq. (2) models the expansion of the negative region $\Omega_t := \{\mathbf{x}, \ \varphi(t, \mathbf{x}) \leq 0\}$ with maximal possible speed, where f_α is the set of possible dynamics (see for instance [3]).

Also, by solving (2) we can recover the minimal time function to reach Ω_0 with the non isotropic (backward) dynamics $-f$:

$$T(\mathbf{x}) := \inf\{t \geq 0, \exists\, \alpha \in L^\infty((-t, 0); \mathcal{A}), \ y_{\mathbf{x}}^\alpha(-t) \in \Omega_0\}$$

Date: Published in SIAM J. Scient. Comput., 33 (2), 923-938.

Key words and phrases. Hamilton-Jacobi-Bellman equations; discontinuous Galerkin methods; level sets; front propagation .

Research supported by ARO grant W911NF-08-1-0520 and NSF grant DMS-0809086.

by simply using the fact that

$$T(\mathbf{x}) = \inf\{t \geq 0,\ \varphi(t, \mathbf{x}) \leq 0\}$$

(where φ is solution of (2) and φ_0 is such that $\Omega_0 := \{\mathbf{x}, \varphi^0(\mathbf{x}) \leq 0\}$). This has been used recently for instance in [5, 4, 6].

It is known since the work of Osher and Sethian [25] that front propagation problems can be solved by using level sets and HJ equations. Level set methods for front propagation in general need "reinitialization" procedures in order to keep a non-vanishing gradient $\nabla\phi(t, \mathbf{x})$ for points $\mathbf{x}$ close to the 0-level set of $\varphi(t, .)$. The quality of the method may also deteriorate for long-time runs because of numerical diffusion, especially when the data have corners, such as a square. It may also have difficulties in handling small fronts with respect to the mesh size.

On the other hand, various numerical methods have been proposed to solve for the viscosity solutions [12, 11] of (1) numerically over the years. In one of the first papers [13], Crandall and Lions studied first order monotone finite difference schemes, which converge to the viscosity solution. Semi-Lagrangian schemes [15], based on the discretization of the dynamic programming principle, are also monotone and thus in general limited to at most first order accuracy [16], however high order modifications have been proposed [14]. Essentially non-oscillatory (ENO) or weighted ENO (WENO) finite difference schemes are popular schemes to solve the Hamilton-Jacobi equation (1), see, e.g. [26, 20, 28]. These finite difference methods work quite efficiently for Cartesian meshes, however on unstructured meshes the scheme is quite complicated [28]. Finite volume methods [2, 21], on the other hand, faces the difficult problem of reconstruction on arbitrary triangulation, see [1] for more discussions.

Alternatively, the Runge-Kutta discontinuous Galerkin (RKDG) method, originally devised to solve the conservation laws, has the advantage of flexibility for arbitrarily unstructured meshes. In [19, 22], DG methods are developed to solve (1) based on the conservation law system satisfied by the derivatives of φ. This has made the algorithm indirect and somewhat complicated. In [9], a DG method for directly solving the Hamilton-Jacobi equation (1) was developed. Extensive numerical tests have been performed for equations with linear and convex nonlinear Hamiltonians. This scheme has provable stability and error estimates for linear equations and demonstrates good convergence to the viscosity solutions for nonlinear equations. However, in entropy violating cells, a correction based on the scheme in [19, 22] is necessary to guarantee stability of the method. In this paper, we modify the DG scheme in [9] to avoid this difficulty. Our aim is to show that the proposed method deals very well with the previous mentioned difficulties on our examples, without the need of "reinitialization" because of the high precision of the method. We focus on front propagation problems (that is, finding the 0-level set of $\varphi(t, .)$) but the proposed method can also be used to compute the value $\varphi(t, .)$ on the whole domain.

In the following section, we present our algorithm. Then, in section 3, we present several numerical results on some typical 2-dimensional problems. We first treat linear problems (equation (2) becomes linear when there is only one control α), and then we treat more general HJB equations of the form (2).

2. Numerical algorithm

In this section, we will discuss a modification of the DG scheme proposed in [9] for the general HJ equation (1). The entropy correction no longer relies on the scheme in [19, 22], resulting in a reduction in computational cost. We will present the method in the context of two-dimensional problems, although the procedure can be easily adapted to one-dimensional and higher multi-dimensional equations.

The two-dimensional Hamilton-Jacobi equation is given by

$$\varphi_t + H(\varphi_x, \varphi_y, x, y) = 0, \qquad \varphi(x, y, 0) = \varphi^0(x, y) \tag{4}$$

As in [9], we assume rectangular domains and cells, although the algorithm described and its designing principle can be readily applied on general triangulations. The main idea is to distinguish the normal and tangential directions along the cell boundaries, and apply the same procedure for each direction.

Suppose the domain of computation is $[a, b] \times [c, d]$. We shall use rectangular meshes defined as

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{N_x+\frac{1}{2}} = b, \qquad c = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \ldots < y_{N_y+\frac{1}{2}} = d \tag{5}$$

and

$$\begin{aligned} &I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}], \quad J_i = [x_{i-1/2}, x_{i+1/2}], \quad K_j = [y_{j-1/2}, y_{j+1/2}] \\ &J_{i+1/2} = [x_i, x_{i+1}], \quad K_{j+1/2} = [y_j, y_{j+1}], \quad i = 1, \ldots N_x, \quad j = 1, \ldots N_y, \end{aligned} \tag{6}$$

where $x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})$, and $y_j = \frac{1}{2}(y_{j-1/2} + y_{j+1/2})$. We define the approximation space as

$$V_h^k = \{v : v|_{I_{i,j}} \in P^k(I_{i,j}),\ i = 1, \ldots N_x, \quad j = 1, \ldots N_y\} \tag{7}$$

where $P^k(I_{i,j})$ denotes all polynomials of degree at most k on $I_{i,j}$.

Let us denote $H_1 = \frac{\partial H}{\partial \varphi_x}$ and $H_2 = \frac{\partial H}{\partial \varphi_y}$. In the cell $I_{i,j}$ we define

$$H_{1,min}^{\varphi_h,i}(y) := \min\left(0, \min_{x \in J_{i+1/2}} H_1(\partial_x \varphi_h(x, y), \overline{\partial_y \varphi_h}(x, y), x_{i+1/2}, y)\right)$$

$$H_{1,max}^{\varphi_h,i}(y) := \max\left(0, \max_{x \in J_{i-1/2}} H_1(\partial_x \varphi_h(x, y), \overline{\partial_y \varphi_h}(x, y), x_{i-1/2}, y)\right)$$

and

$$H_{2,min}^{\varphi_h,j}(x) := \min\left(0, \min_{y \in K_{j+1/2}} H_2(\overline{\partial_x \varphi_h}(x, y), \partial_y \varphi_h(x, y), x, y_{j+1/2})\right)$$

$$H_{2,max}^{\varphi_h,j}(x) := \max\left(0, \max_{y \in K_{j-1/2}} H_2(\overline{\partial_x \varphi_h}(x, y), \partial_y \varphi_h(x, y), x, y_{j-1/2})\right)$$

where we have used the notations

$$\overline{\partial_x \varphi_h} = \frac{1}{2}\left((\partial_x \varphi_h)^+ + (\partial_x \varphi_h)^-\right), \qquad \overline{\partial_y \varphi_h} = \frac{1}{2}\left((\partial_y \varphi_h)^+ + (\partial_y \varphi_h)^-\right).$$

Here and below, the superscript $+$ is used to denote the right (in x-direction) or top (in y-direction) limit of the function, whereas, the superscript $-$ is used to denote the left (in x-direction) or bottom (in y-direction) limit of the function.

Then the scheme introduced in [9] is: find $\varphi_h(x,t) \in V_h^k$, such that

$$\begin{aligned} 0 = \int_{I_{i,j}} &(\partial_t \varphi_h(x,y,t) + H(\partial_x \varphi_h(x,y,t), \partial_y \varphi_h(x,y,t), x, y)) v_h(x,y) dxdy \\ &+ \int_{K_j} H_{1,min}^{\varphi_h,i}(y)\ [\varphi_h](x_{i+\frac{1}{2}}, y) v_h(x_{i+\frac{1}{2}}^-, y) dy \\ &+ \int_{K_j} H_{1,max}^{\varphi_h,i}(y)\ [\varphi_h](x_{i-\frac{1}{2}}, y) v_h(x_{i-\frac{1}{2}}^+, y) dy \\ &+ \int_{J_i} H_{2,min}^{\varphi_h,j}(x)\ [\varphi_h](x, y_{j+\frac{1}{2}}) v_h(x, y_{j+\frac{1}{2}}^-) dx \\ &+ \int_{J_i} H_{2,max}^{\varphi_h,j}(x)\ [\varphi_h](x, y_{j-\frac{1}{2}}) v_h(x, y_{j-\frac{1}{2}}^+) dx \end{aligned} \tag{8}$$

holds for any $v_h \in V_h^k$, where we have denoted $[\varphi_h](x_{i+\frac{1}{2}}, y) := \varphi_h(x_{i+\frac{1}{2}}^+, y) - \varphi_h(x_{i+\frac{1}{2}}^-, y)$ and $[\varphi_h](x, y_{j+\frac{1}{2}}) := \varphi_h(x, y_{j+\frac{1}{2}}^+) - \varphi_h(x, y_{j+\frac{1}{2}}^-)$.

Because the solution is discontinuous at interfaces of cells, reconstructions are needed. Along the normal direction of the interface, we would use the L^2 reconstructed information of the partial derivatives as in the one-dimensional case. Tangential to the interface, the average of the partial derivatives from the two neighboring cells is used. The reconstruction process is described in [9].

For general nonlinear equations, a suitable entropy correction is necessary to guarantee the stability of the scheme. The criteria for the violation of entropy condition are simple and are described below. We say the entropy condition is violated at $x = x_{i\pm\frac{1}{2}}$, if

$$H_1(\partial_x \varphi_h^-(x_{i\pm\frac{1}{2}}, y_j), \partial_y \varphi_h^-(x_{i\pm\frac{1}{2}}, y_j), x_{i\pm\frac{1}{2}}, y_j) < 0$$

and

$$H_1(\partial_x \varphi_h^+(x_{i\pm\frac{1}{2}}, y_j), \partial_y \varphi_h^+(x_{i\pm\frac{1}{2}}, y_j), x_{i\pm\frac{1}{2}}, y_j) > 0.$$

Similarly, the entropy condition is violated at $y = y_{j\pm\frac{1}{2}}$, if

$$H_2(\partial_x \varphi_h^-(x_i, y_{j\pm\frac{1}{2}}), \partial_y \varphi_h^-(x_i, y_{j\pm\frac{1}{2}}), x_i, y_{j\pm\frac{1}{2}}) < 0$$

and

$$H_2(\partial_x \varphi_h^+(x_i, y_{j\pm\frac{1}{2}}), \partial_y \varphi_h^+(x_i, y_{j\pm\frac{1}{2}}), x_i, y_{j\pm\frac{1}{2}}) > 0.$$

A drawback of the simple entropy correction described below is that accuracy will degenerate to at most second order when the entropy correction is performed. In order to avoid unnecessary corrections, we add one additional constraint. We will only perform the entropy fix if

$$H(\partial_x \varphi_h(x_i, y_j), \partial_y \varphi_h(x_i, y_j), x_i, y_j)) > \epsilon,$$

where ϵ is a small number which is taken as 10^{-3} in our numerical experiments. This threshold value could be problem dependent and we can only state that $\epsilon = 10^{-3}$ works well for all our test cases.

In [9], the entropy fix was proposed as follows.

(1) For each cell $I_{i,j}$, determine if it is a potentially entropy violating cell. If the cell $I_{i,j}$ is marked as a potentially entropy violating cell, then use Step 2 below to update φ_h in this cell; otherwise, update φ_h by (8).
(2) Update φ_h by the DG method of Hu and Shu [19] as reinterpreted by Li and Shu [22], namely, recover $\partial_x\varphi_h$ and $\partial_y\varphi_h$ by taking the derivatives of φ_h, then compute $\partial_t(\partial_x\varphi_h)$ and $\partial_t(\partial_y\varphi_h)$ by the usual DG method for the conservation laws satisfied by $\partial_x\varphi_h$ and $\partial_y\varphi_h$ in a locally curl-free discontinuous Galerkin space. This will determine φ_h up to a constant. The missing constant is obtained by requiring

$$\int_{I_{i,j}} (\partial_t\varphi_h + H(\partial_x\varphi_h, \partial_y\varphi_h, x, y))dxdy = 0. \tag{9}$$

Although the above entropy fix is performed at only a few isolated cells in each time step [9], it will have a higher demand for computational cost and storage, because the correction step relies on rewriting one HJ equation into a system of equations for the solution's derivatives. Here and below, we present a new entropy fix to scheme (8). This entropy fix is inspired by the finite difference scheme proposed in [8]. The main motivation comes from the linear equations. As shown in [9], scheme (8) will reduce to a standard DG scheme for conservations laws with the upwind flux and a source term in that case. In [8], the N-Bee scheme for linear equations with variable coefficients is considered, and the authors discussed various cases depending on the signs of the coefficients. We can identify that the entropy fix described below will reduce to one of the special cases of the N-Bee scheme corresponding to the specific requirement of the entropy violation.

For the cell $I_{i,j}$, if the entropy condition is violated, we cast it into four categories.

(1) If the entropy condition is violated at the right boundary $x = x_{i+\frac{1}{2}}$, and $\varphi_h^+(x_{i+\frac{1}{2}}, y)$ is not in the interval between $\varphi_h(x_i, y)$ and $\varphi_h(x_{i+1}, y)$, then replace the term $[\varphi_h](x_{i+\frac{1}{2}}, y)$ in (8) by $(\varphi_h(x_{i+1}, y) + \varphi_h(x_i, y))/2 - \varphi_h^-(x_{i+\frac{1}{2}}, y)$. In practice, we only need to do it for the y values that are located at the Gaussian quadrature points.
(2) If the entropy condition is violated at the left boundary $x = x_{i-\frac{1}{2}}$, and $\varphi_h^-(x_{i-\frac{1}{2}}, y)$ is not in the interval between $\varphi_h(x_{i-1}, y)$ and $\varphi_h(x_i, y)$, then replace the term $[\varphi_h](x_{i-\frac{1}{2}}, y)$ in (8) by $\varphi_h^+(x_{i-\frac{1}{2}}, y) - (\varphi_h(x_{i-1}, y) + \varphi_h(x_i, y))/2$. In practice, we only need to do it for the y values that are located at the Gaussian quadrature points.
(3) If the entropy condition is violated at the top boundary $y = y_{j+\frac{1}{2}}$, and $\varphi_h^+(x, y_{j+\frac{1}{2}})$ is not in the interval between $\varphi_h(x, y_j)$ and $\varphi_h(x, y_{j+1})$, then replace the term $[\varphi_h](x, y_{j+\frac{1}{2}})$ in (8) by $(\varphi_h(x, y_{j+1}) + \varphi_h(x, y_j))/2 - \varphi_h^-(x, y_{j+\frac{1}{2}})$. In practice, we only need to do it for the x values that are located at the Gaussian quadrature points.
(4) If the entropy condition is violated at the top boundary $y = y_{j-\frac{1}{2}}$, and $\varphi_h^-(x, y_{j-\frac{1}{2}})$ is not in the interval between $\varphi_h(x, y_{j-1})$ and $\varphi_h(x, y_j)$, then replace the term $[\varphi_h](x, y_{j-\frac{1}{2}})$ in (8) by $\varphi_h^+(x, y_{j+\frac{1}{2}}) - (\varphi_h(x, y_j) + \varphi_h(x, y_{j-1}))/2$. In practice, we only need to do it for the x values that are located at the Gaussian quadrature points.

This modification incurs almost no additional computational cost and little change to the code. Note that we were checking the values at edge quadrature points only since this saves computational cost and works for all our numerical tests.

The method of line ODE obtained from (8) and its corrections is then solved by a total variation diminishing (TVD) high-order Runge-Kutta time discretization as usual, see [27], also [17, 18] for details.

3. Numerical results

In this section, we present numerical results to validate our scheme. In all tests, the DG scheme is coupled with a third-order TVD Runge-Kutta method as in [9]. This accuracy in time is enough for our application.

For testing the evolution of fronts, in some cases we shall compute the Hausdorff distance between the approximated front and the exact front. The Hausdorff distance between two sets A, B is defined by

$$d_H(A,B) := \max(\max_{a\in A} d(a,B), \max_{b\in B} d(b,A)).$$

3.1. **Linear advection examples.**

In this subsection, we will consider linear problems given by

$$\begin{cases} \varphi_t + f(t,x,y)\cdot\nabla\varphi = 0 \quad (x,y)\in\Omega,\ t\in[0,T] \\ \varphi(0,x,y) = \varphi^0(x,y) \end{cases} \tag{10}$$

with $\Omega\subset\mathbb{R}^2$. For this type of equations, the scheme is equivalent to the DG scheme for conservation laws with source terms and no entropy correction is necessary [9].

Example I.1: We consider a rotation test, where

$$f(\mathbf{x}) := 2\pi(-y,x)^T,$$

and we impose a "circle" initial data

$$\varphi^0(x,y) := \min(\|\mathbf{x}-\mathbf{x}_A\|_2 - r, r), \qquad \mathbf{x}_A := (0,1),\ \ r = 0.5. \tag{11}$$

The solution will rotate around the origin with a period of 1, and $\{\mathbf{x}\in\mathbb{R}^2,\ \varphi^0(\mathbf{x}) = 0\}$ corresponds to a circle centered in $\mathbf{x}_A$ and of radius r.

In Table 1, we show the *local error*, that is, only considering the mesh cells $(\mathbf{x}_I)$ such that $|\varphi(t,\mathbf{x}_I)|\le\eta$ (with the threshold $\eta = 0.15$ for this example). This is the error in a neighborhood of the 0-level set. We also compute an estimate of the Hausdorff distance between the exact 0-level set and the approximate 0-level set. Because the solution is smooth around the 0-level set, we observe third order behavior for the Hausdorff distance, as well as for the local error.

The performance of the scheme is good in particular with respect to long time, since the error does not deteriorate much between times $t = 1$ and $t = 10$. This property has been reported for the standard DG applied to conservation laws [10].

Remark 3.1. *We also observe numerically global convergence of the scheme with an order of approximately one, since the solution is only Lipschitz continuous but not continuously differentiable.*

TABLE 1. (Example I.1) Local error and Hausdorff distance, $t = 1$ and $t = 10$, with P^2.

$t = 1$

$N_x = N_y$	$\Delta x = \Delta y$	L^1-error	order	L^2-error	order	L^∞-error	order	d_H	order
10	0.5	1.03e-02	-	1.34e-02	-	3.84e-02	-	3.29e-02	-
20	0.25	4.27e-03	1.27	5.36e-03	1.32	1.76e-02	1.13	9.86e-03	1.74
40	0.125	4.28e-04	3.32	5.66e-04	3.24	2.90e-03	2.60	1.64e-03	2.59
80	0.0675	4.76e-05	3.17	6.22e-05	3.19	2.55e-04	3.51	1.33e-04	3.63

$t = 10$

$N_x = N_y$	$\Delta x = \Delta y$	L^1-error	order	L^2-error	order	L^∞-error	order	d_H	order
10	0.5	4.66e-02	-	5.62e-02	-	1.30e-01	-	1.17e-01	-
20	0.25	8.59e-03	2.44	1.01e-02	2.48	2.33e-02	2.48	1.19e-02	3.30
40	0.125	1.65e-03	2.38	1.99e-03	2.34	6.09e-03	1.93	3.33e-03	1.83
80	0.0675	2.31e-04	2.84	2.91e-04	2.78	7.89e-04	2.95	2.73e-04	3.61

Example I.2: We consider the same rotation test as above, but with a "square" initial data

$$\varphi^0(x, y) := \min(\|\mathbf{x} - \mathbf{x}_A\|_\infty - r, r), \qquad \mathbf{x}_A := (0, 1), \quad r = 0.5. \tag{12}$$

Here $\{\mathbf{x} \in \mathbb{R}^2,\ \varphi(\mathbf{x}) = 0\}$ corresponds to a square centered in $\mathbf{x}_A$ and of side length $2r$. The computational domain is $[-2.5, 2.5]^2$. Results are shown in Fig. 1 at times $t = 1$ and $t = 10$. Again, we observe a good performance of the scheme with respect to long time.

We numerically observe a local first-order convergence on this example, see Table 2. The degeneracy of local errors is due to the non-smoothness of the solution near the 0-level set.

TABLE 2. (Example I.2) Local error at time $t = 1$ and $t = 10$, with P^3

$t = 1$

N_x	Δx	L^1-error	order	L^2-error	order	L^∞-error	order	d_H	order
10	0.5	2.26e-02	-	2.62e-02	-	8.44e-02	-	5.92e-02	-
20	0.25	8.60e-03	1.40	1.04e-02	1.33	3.62e-02	1.22	3.12e-02	0.92
40	0.125	2.86e-03	1.59	4.13e-03	1.33	2.04e-02	0.82	1.81e-02	0.79
80	0.0675	8.84e-04	1.69	1.67e-03	1.31	1.08e-02	0.93	1.00e-02	0.85

$t = 10$

N_x	Δx	L^1-error	order	L^2-error	order	L^∞-error	order	d_H	order
10	0.5	4.64e-02	-	5.21e-02	-	1.38e-01	-	8.75e-02	-
20	0.25	1.32e-02	1.82	1.50e-02	1.79	5.25e-02	1.39	3.73e-02	1.23
40	0.125	5.07e-03	1.38	6.60e-03	1.19	2.85e-02	0.88	2.57e-02	0.54
80	0.0675	1.61e-03	1.65	2.63e-03	1.33	1.49e-02	0.93	1.43e-02	0.85

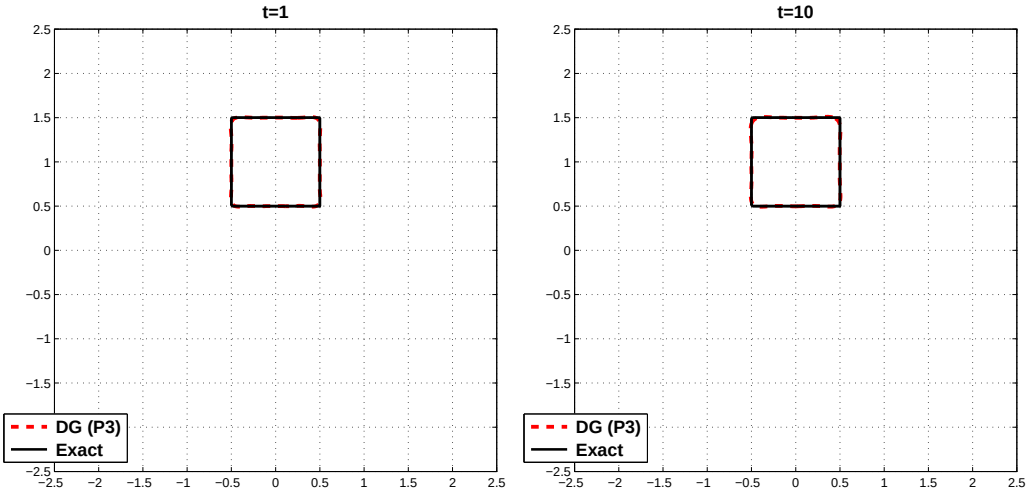

FIGURE 1. (Example I.2) $t = 1$ (left), and $t = 10$ (right), with P^3 and $N_x = N_y = 40$

Example I.3: We consider

$$f(t,x,y) := sign\left(\frac{T}{2} - t\right)\begin{pmatrix} -2\pi\, y \\ 2\pi\, x \end{pmatrix} \max(1 - \|\mathbf{x}\|_2, 0).$$

where $\|\mathbf{x}\|_2 := \sqrt{x^2+y^2}$ and with a Lipschitz continuous initial data φ:

$$\varphi^0(x,y) = \min(\max(y,-1),1). \tag{13}$$

The function φ^0 has a 0-level set which is the x axis: $\{\mathbf{x} = (x,y) \in \mathbb{R}^2 \mid y = 0\}$. The exact solution is known.[1]

In this example the front evolves up to time $t = T/2$ then it comes back to the initial data at time $t = T$. $T/2$ represents the number of turns.

Computations have been done up to time $T = 10$ (see Fig. 2), on the domain $(x,y) \in [-1.2, 1.2]^2$, on a 24×24 mesh, with piecewise P^4 in space. This represents the solution obtained by high order polynomials computed on a rather coarse mesh. We obtain very good resolution, except at places where f is nonsmooth, i.e. $||\mathbf{x}||_2 = 1$, where small oscillation occurs.

Example I.4: Here we consider "thin target" examples. We first consider the advection case $f = (1,1)^T$. The initial condition is $\varphi^0 := \varphi^0_r$ where $r = \Delta x$ is the mesh size, and

$$\varphi^0_r(\mathbf{x}) := \min(r, \|\mathbf{x}\|_\infty - r).$$

The set $\{x, \varphi^0(x) = 0\}$ thus corresponds to a small square box, of length $2\Delta x$.

Computations are done on $[-1,\ 1]^2$ with periodic boundary conditions, on a 20×20 mesh, with P^3 polynomials, and for $t = 2$ (one period) and $t = 10$ (five periods). Results are shown in Fig. 3.

[1]It is given by $\varphi(t,\mathbf{x}) := \varphi^0(R(-2\pi a(\mathbf{x})\min(t, T-t))\ \mathbf{x})$ where $R(\theta) := \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$ and $a(\mathbf{x}) := \max(1 - \|\mathbf{x}\|_2, 0)$.

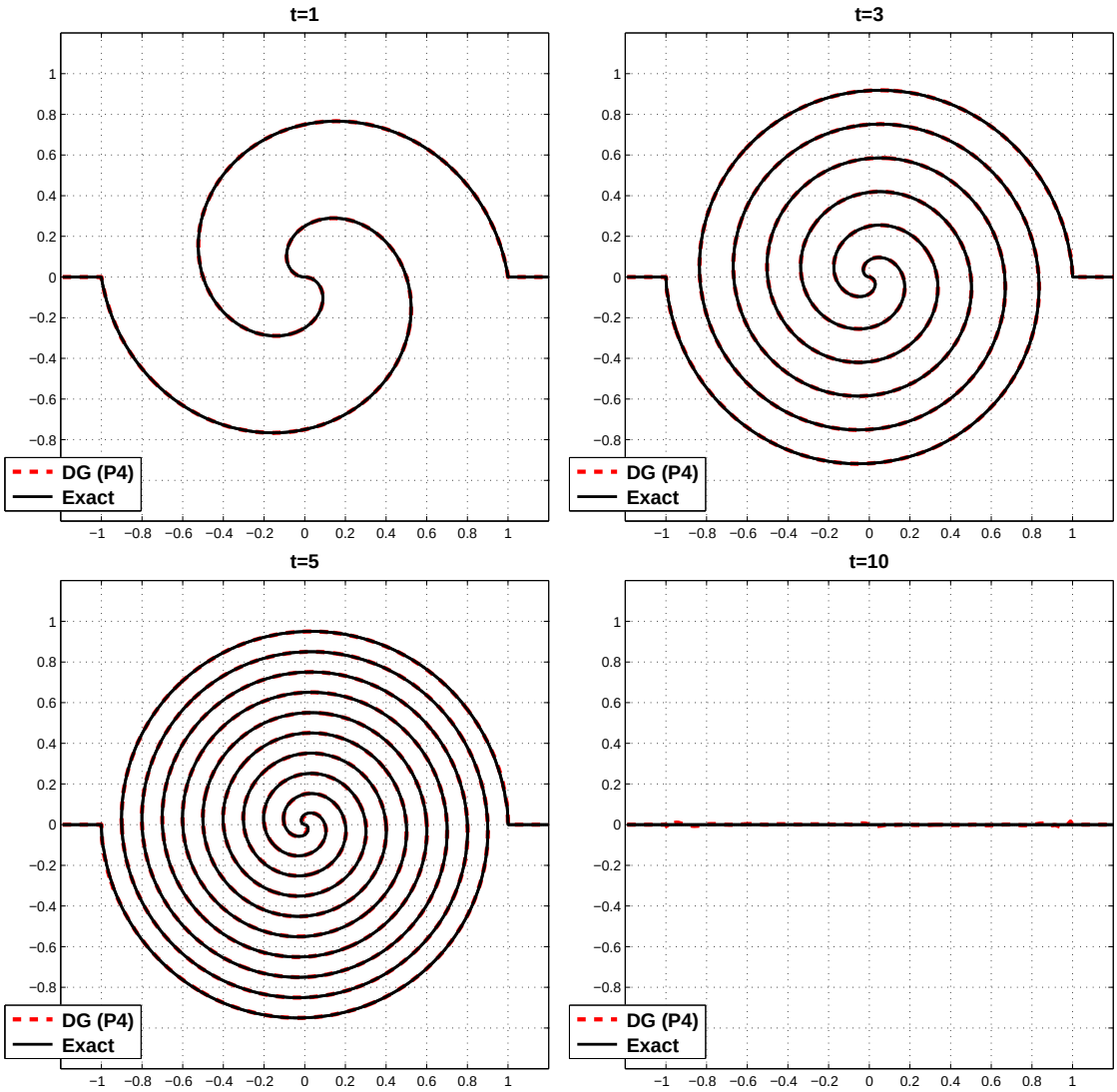

FIGURE 2. (Example I.3) Plots at times $t = 1$, $t = 3$, $t = 5$ and $t = 10$ (return to initial data), with P^4 and 24×24 mesh cells.

Secondly, we consider the case of $f(x, y) = 2\pi(-y, x)^T$ (rotation) with a similar small square initial condition:

$$\varphi_r^0(\mathbf{x}) := \min(r, \|\mathbf{x} - (0.5, 0)\|_\infty - r).$$

where $r = \Delta x$ (the mesh size).

Computations are done on the domain $[-1, 1]^2$ with a piecewise P^3 polynomial on a 20×20 mesh, at times $t = 1$ (one period) and $t = 5$ (five periods). Results are given in Fig. 4.

We obtain in both cases impressive long time resolution for thin targets that are relatively small compared to the mesh size.

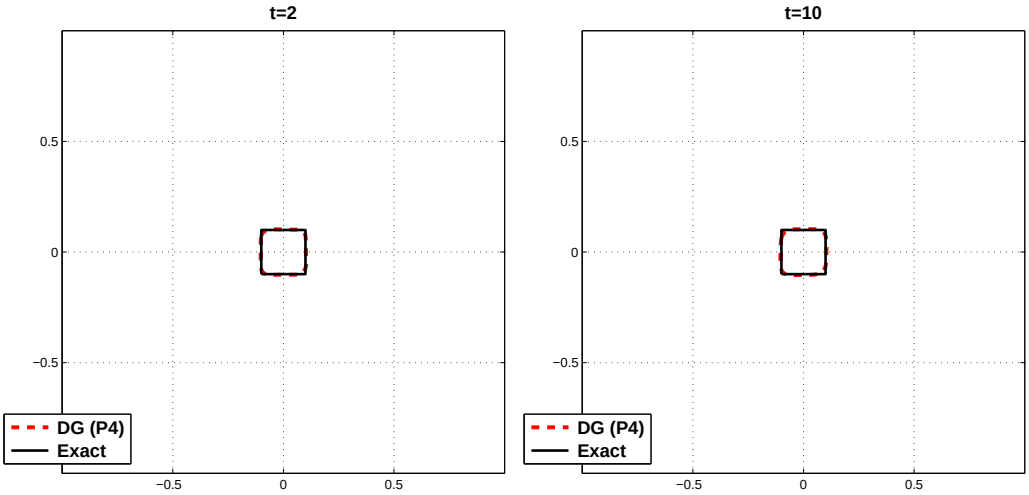

FIGURE 3. Example I.4, advection of a small target, $t = 2$ (left) and $t = 10$ (right), P^4 with 20×20 mesh cells.

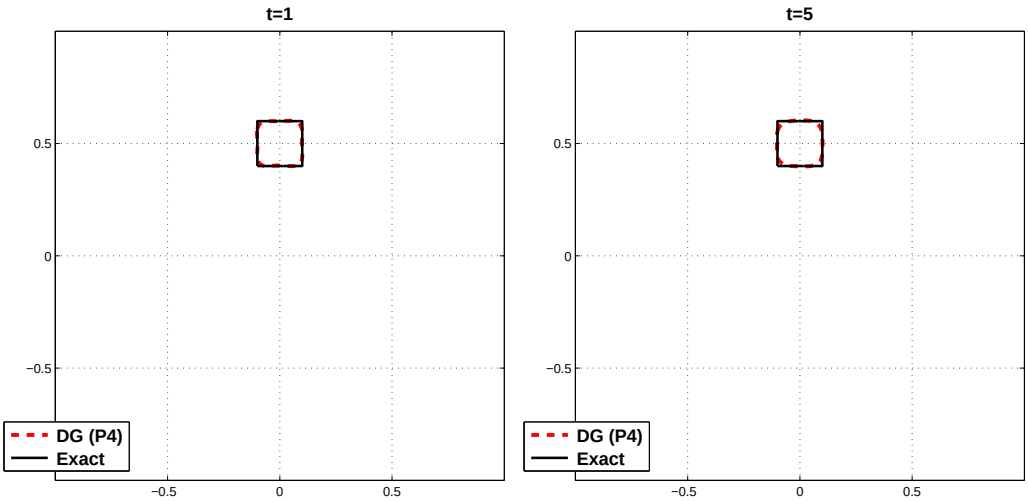

FIGURE 4. Example I.4, rotation of a small target, $t = 1$ (left) and $t = 5$ (right), P^4 with 20×20 cells.

3.2. **HJB examples.** In this subsection, we will consider the general Hamilton-Jacobi equation (1) and (4).

Example II.1: (reachability problem) We consider here

$$\varphi_t + \max(0, f \cdot \nabla \varphi) = 0,$$

with the initial data $\varphi^0 := \min(r_0, \|\mathbf{x} - \mathbf{x}_A\|_2 - r_0)$ ($r_0 = 0.25$, $\mathbf{x}_A = (\frac{1}{2}, 0)^T$). The region $C := \{\varphi^0(\mathbf{x}) \leq 0\}$ is the disk centered at A and of radius r_0. The dynamics is

$$f(t, x, y) := 2\pi \begin{pmatrix} -y \\ x \end{pmatrix}.$$

Remark 3.2. *In this example the negative region* $\{\mathbf{x}, \varphi(t, \mathbf{x}) \leq 0\}$ *corresponds to the points* $\mathbf{x}$ *from which we can find a trajectory solution of* $\dot{y} = f(y)$, $y(0) = \mathbf{x}$ *that can reach the "target"* C *before some given time* t *(see for instance* [24]*).*

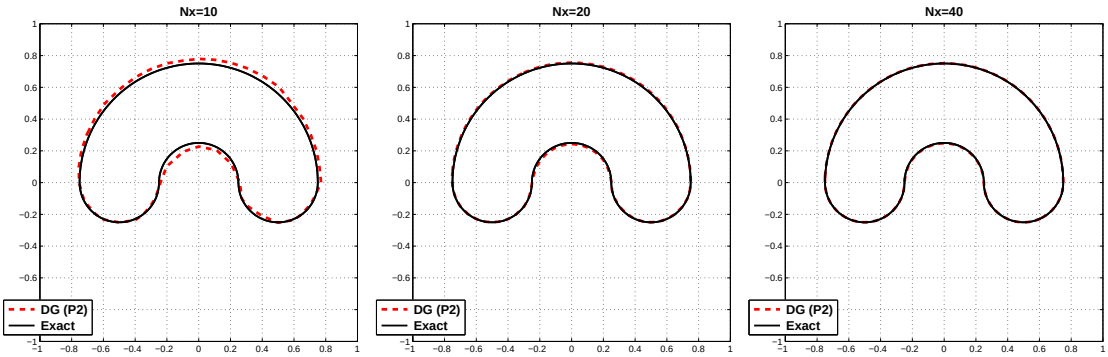

FIGURE 5. Example II.1, $t = 0.5$, with (P^2) and $N_x = 10$, 20 and 40.

In this example, there is no need to use the entropy fix. Results are given in Figure 5 for three different mesh sizes. The local errors are also computed in the neighborhood of the front such that $|\varphi(t, \mathbf{x})| < 0.15$ (see Table 3). We observe a local error approximately of order 2. Here the degeneracy of the order comes from the fact that the solution near the front $\varphi = 0$ is not twice continuously differentiable. This is true also for the Hausdorff distance.

TABLE 3. Example II.1, DG (P^2). "Local" errors.

$N_x = N_y$	$\Delta x = \Delta y$	L^1-error	order	L^2-error	order	L^∞-error	order	d_H	order
10	0.2	2.70e-02	-	2.92e-02	-	6.03e-02	-	3.67e-02	-
20	0.1	7.63e-03	1.82	9.42e-03	1.63	3.18e-02	0.92	8.99e-03	2.03
40	0.05	1.84e-03	2.06	2.26e-03	2.06	1.11e-02	1.52	2.54e-03	1.82
80	0.025	4.49e-04	2.03	5.27e-04	2.10	1.55e-03	2.84	7.70e-04	1.72

Example II.2: We consider

$$\varphi_t(t, x) + \varphi_x(t, x) + |\varphi_y(t, x)| = 0, \quad t \in [0, T], \ x = (x, y) \in \mathbb{R}^2. \tag{14a}$$

$$\varphi(0, x) = \varphi_r^0(x). \tag{14b}$$

For $r > 0$, the initial condition is given by

$$\varphi_r^0(x) := \min(\|x\|_\infty - r, r). \tag{15}$$

Two types of tests have been realized:

- $r = 0.1$: "large target" case. The initial condition is a centered square of length $2r$.
- $r = 0.0$: "thin target" case (a limit case when $r \to 0+$). Numerically, we manage this by putting $(0,0)$ at the interface of cell boundaries and with $\Delta x = \Delta y$, and we choose
$$\varphi^0(x) := \min((p-1)r_1,\ p(\|x\|_\infty - r_1)),$$
with $r_1 = \Delta x/p$ and $p = 10$ (Hence $\{x, \varphi^0(x) = 0\}$ is the small square box $\|x\|_\infty = r_1 = \Delta x/p$).

We first treat the case $r = 0.1$. In Fig. 6 we have represented a computation without entropy fix (left) and with an entropy fix (right). The cells where an entropy fix is used (at the time of the current computation) are marked with a star.

We see that the entropy modification is necessary, otherwise instabilities develop and we get a wrong approximation.

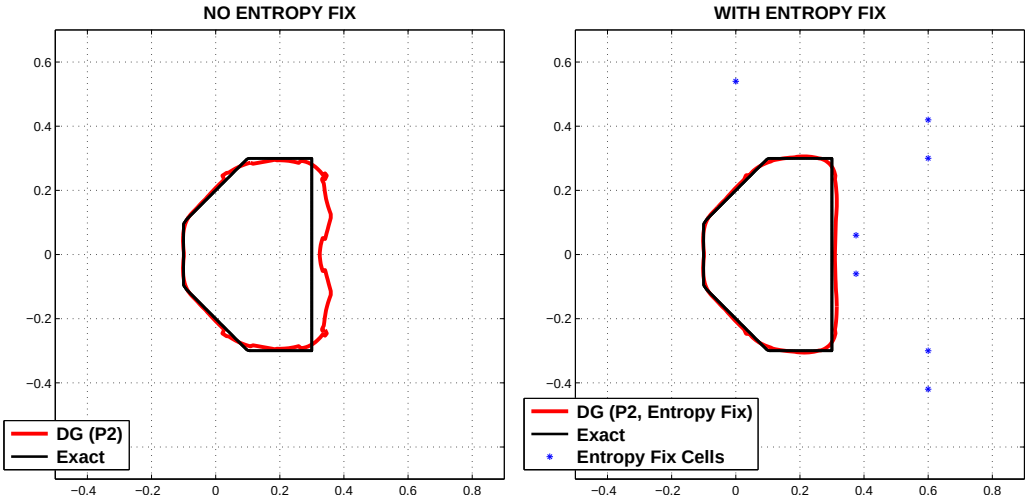

FIGURE 6. (Example II.2) $t = 0.2$, no entropy fix (left) and with entropy fix (right), with 15×25 mesh cells and P^2 polynomials.

We now turn to the case $r = 0$. Results are given in Figure 7. (The exact solution satisfies $\{x,\ \varphi(t,x) \le 0\} = \{(x,y),\ |y| \le x \le t\}$.) This case is numerically more difficult because the initial data have strong variations near the point $(0,0)$, so that the initial front is almost one point. The scheme resolves the case very well.

Example II.3: The HJB equation we consider is

$$\varphi_t(t,x) - y\varphi_x(t,x) + |\varphi_y(t,x)| = 0, \quad t \in [0,T],\ (x,y) \in \mathbb{R}^2. \tag{16a}$$

$$\varphi(0,x) = \varphi^0(x). \tag{16b}$$

As in the previous example (case $r = 0$) the initial data should be such that $\varphi^0(\mathbf{x}) < 0$ if $x = (0,0)$, and $\varphi^0(\mathbf{x}) > 0$ otherwise.

Remark 3.3. *This example comes from the control problem to reach* $(0,0)$ *from a given point* $\boldsymbol{x}_0 \in \mathbb{R}^2$ *with dynamic* $\ddot{y} = u \in [-1,1]$. *The analytic solution is known for the front*

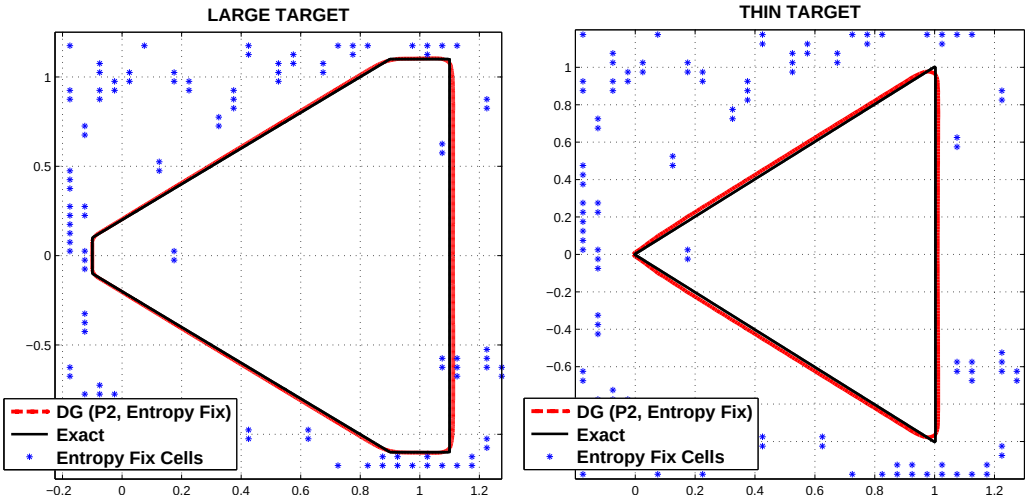

FIGURE 7. (Example II.2) $t = 1.0$, $(N_x, N_y) = (30, 48)$ with entropy fix, $r = 0.1$ (left) and $r = 0.0$ (right)

localization: for any $t \geq 0$, we have

$$\{\boldsymbol{x},\ \varphi(t,\boldsymbol{x}) = 0\} = \{\boldsymbol{x} = (x,y),\ |y| \leq t,\ x = \frac{1}{4}(|y| - t)^2 - \frac{t^2}{2}\}.$$

This HJB equation is not easy to approximate by the usual level set methods with a coarse mesh (see [23, 7]).

Here, we choose φ as in Example II.2 for $r = 0$ and mesh size $N_x = N_y = 44$, with P^2. We obtain nice numerical results as shown in Figure 8 (left). Entropy fix is needed in this example.

For comparison, we have also plotted the result obtained with a traditional level set solver, on a grid of 401^2 mesh cells, based on a Lax-Friedrich scheme following [26] (RK2 in time and ENO2 for spatial derivatives) and using Mitchell's Toolbox [23]. In this case the data is initialized with a small square box with $r = 0.005$ and $\varphi = \varphi_r$ as in (15). We can see that the resolution is not as good as the DG solution even though the mesh is much more refined.

Example II.4: We solve

$$\varphi_t + 2\pi \begin{pmatrix} -y \\ x \end{pmatrix} \cdot \nabla\varphi + sign(\sin(2\pi t)) \left| \begin{pmatrix} -\sin(2\pi t) \\ \cos(2\pi t) \end{pmatrix} \cdot \nabla\varphi \right| = 0$$

and φ^0 corresponds to a "square box":

$$\varphi^0(\mathbf{x}) := \min(r_0, \|\mathbf{x} - \mathbf{x}_A\|_\infty - r_0)$$

with $\mathbf{x}_A = (1,0)^T$ and $r_0 = 0.2$. The exact solution is 2-periodic, and for odd integer times ($t = 2n+1$, $n \in \mathbb{N}$), the square box is transformed into a rectangular box

$$\Omega := [1 - r_0, 1 + r_0] \times [-(1 + r_0), (1 + r_0)].$$

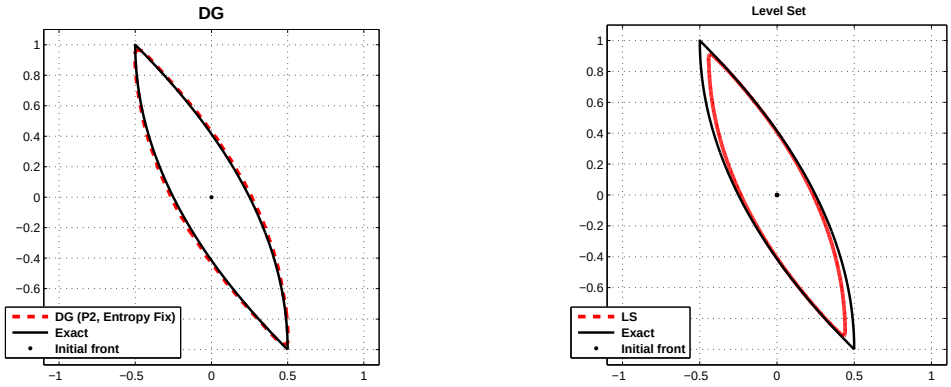

FIGURE 8. (Example II.3) Comparison at time $t = 1.0$: DG scheme with $N_x = N_y = 44$, P^2 (left) and traditional level set method using a second order Lax-Friedrich type scheme (right).

We computed on the domain $[-2.5, 2.5]^2$ with a piecewise P^2 polynomial on a 50×50 mesh ($\Delta x = \Delta y = 0.1$). The results are shown in Fig. 9. For this example there is no need for the entropy fix.

Example II.5: Eikonal equation with rotation. We consider the following HJ equation:

$$\varphi_t + \|\nabla\varphi\| + 4\pi \begin{pmatrix} -y \\ x \end{pmatrix} \cdot \nabla\varphi = 0$$

with an initial data coding two circles:

$$\varphi^0(\mathbf{x}) := \min(r_0, \|\mathbf{x} - \mathbf{x}_A\|_2 - r_0, \|\mathbf{x} - \mathbf{x}_B\|_2 - r_0),$$

$\mathbf{x}_A = (1, 0)^T$, $\mathbf{x}_B = (-1, 0)^T$, $r_0 = 0.5$. Computations are done in the domain $\Omega = [-3, 3]^2$ with periodic boundary conditions. Note that here the boundary conditions do not really change the computations since we expect to have $\varphi(t, x) \simeq r_0$ near the boundary.

Results are shown in Fig. 10 for times $t = 0$ up to $t = 1$. The initial two circles expand and rotate in the same time. They touch at time $t = 0.5$ (after one turn), and then merge. There is no need of the entropy fix for this example. We can observe a very nice merging of the fronts.

Example II.6: We consider the evolution with

$$\varphi_t + \max(|\varphi_x|, |\varphi_y|) = 0.$$

The initial condition is defined by

$$\varphi^0(x, y) := \min(r_0, d - r_0)$$

where $d = \min(d_1, d_2, d_3, d_4)$ and $d_i = \|R(\mathbf{x} - \mathbf{x}_{A_i})\|_\infty$, where A_1, A_2, A_3, A_4 are the four points $\{(1, 1), (1, -1), (-1, 1), (-1, -1)\}$ and R is the rotation of angle $\pi/2$ and centered at the origin. The 0-level set of φ is showed in Fig. 11. Entropy fix is needed for this example. A good merging of the fronts is observed.

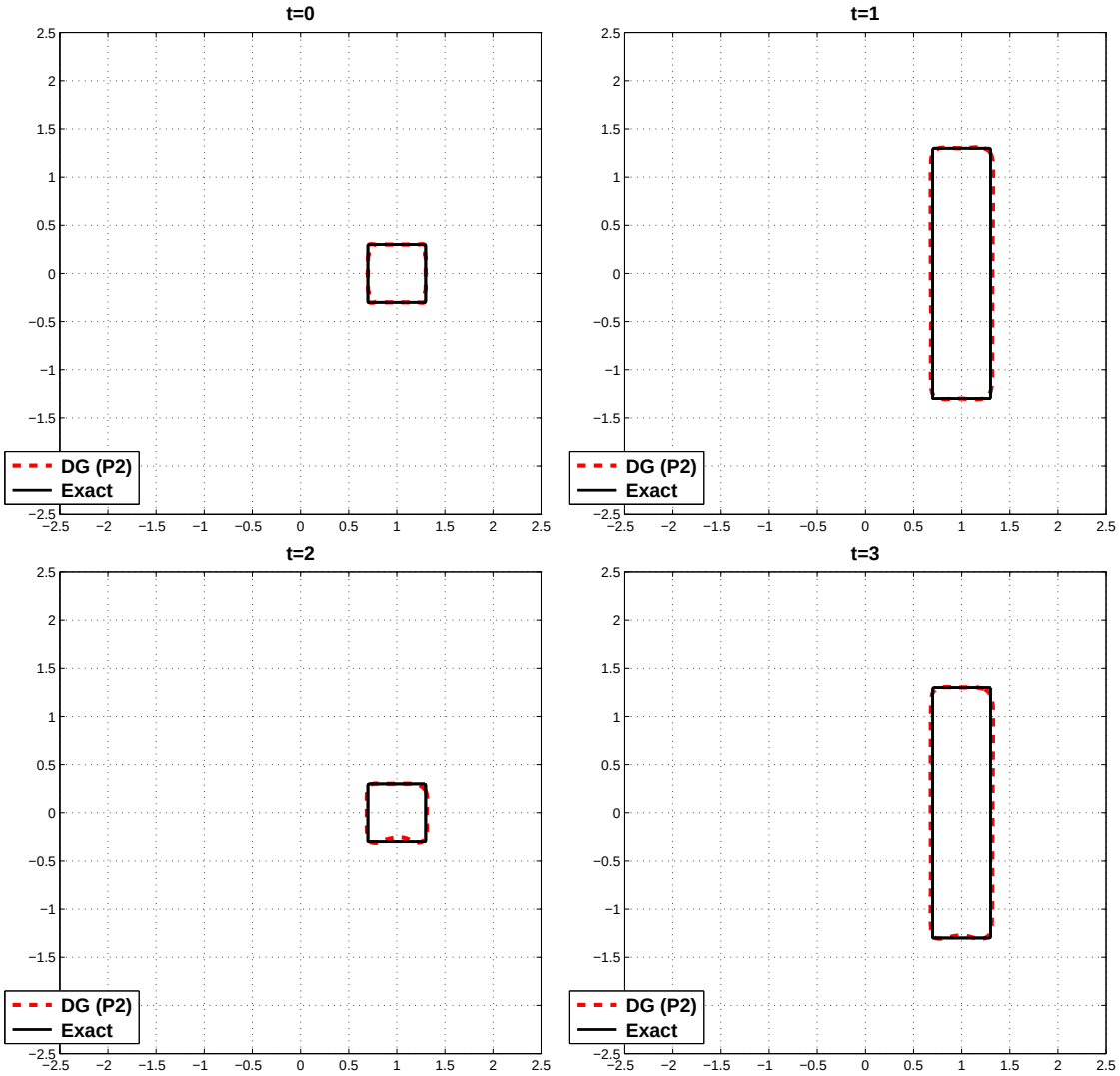

FIGURE 9. Example II.4, at times $t = 0$ (initial data) and $t = 1$, $t = 2$, $t = 3$, with P^2, 50×50 mesh cells.

4. Conclusion

We have proposed a new DG method for the treatment of HJ(B) equations that arise from optimal control problems and that also model a type of front propagation problems with non-isotropic dynamic. We have shown numerically that we are able to recover high order of accuracy for the front localization if the solution is smooth there. More importantly, we have observed very good long time behavior. The resolution deteriorates very slowly with time, and the high order DG method has very small numerical diffusion. Therefore, we do not need any reinitialization of the front. This method is also able to deal with thin fronts with respect to the mesh size.

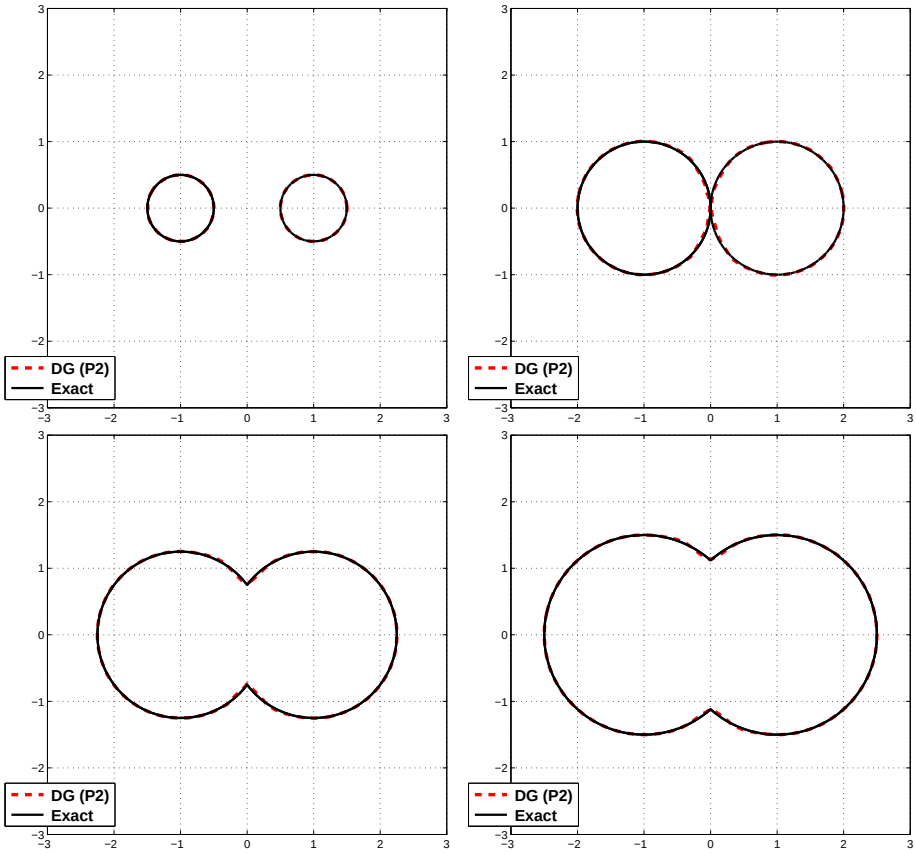

FIGURE 10. (Example II.5) $t = 0$, $t = 0.5$, $t = 0.75$, $t = 1$, using $N_x = N_y = 20$ and no entropy fix.

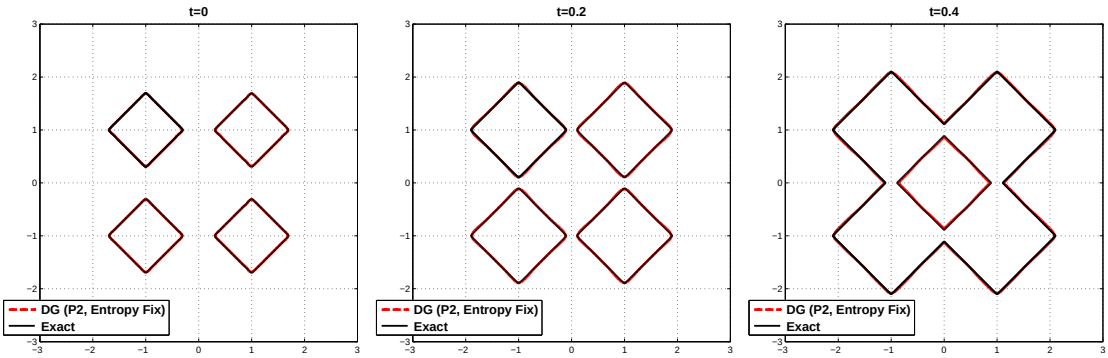

FIGURE 11. (Example II.6) $t = 0$, $t = 0.2$, $t = 0.4$, using $N_x = N_y = 40$ with entropy fix.

In future work, we expect to propose a narrow band method adapted to this DG scheme. We also intend to treat front propagation problems in the presence of obstacles (forbidden region for the trajectories of (3)).

References

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *J. Comput. Phys.*, 114:45–58, 1994.

[2] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Comm. Pure Appl. Math.*, 49:1339–1373, 1996.

[3] M. Bardi and I. Capuzzo-Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Systems and Control: Foundations and Applications. Birkhäuser, Boston, 1997.

[4] O. Bokanowski, A. Briani, and H. Zidani. Minimum time control problems for non autonomous differential equations. *Systems & Control Letters*, 58(10-11):742–746, 2009.

[5] O. Bokanowski, E. Cristiani, J. Laurent-Varin, and H. Zidani. Hamilton-Jacobi-Bellman approach for the climbing problem. *Research report*.

[6] O. Bokanowski, N. Foradel, and H. Zidani. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control Optim.*, 48(7):4292–4316, 2010.

[7] O. Bokanowski, S. Martin, R. Munos, and H. Zidani. An anti-diffusive scheme for viability problems. *Appl. Numer. Math.*, 56(Issue 9, in Numerical Methods for Viscosity Solutions and Applications):1135–1254, 2006.

[8] O. Bokanowski and H. Zidani. Anti-diffusive schemes for linear advection and application to Hamilton-Jacobi-Bellman equations. *J. Sci. Comput.*, 30(1):1–33, 2007.

[9] Y. Cheng and C.-W. Shu. A discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *J. Comput. Phys.*, 223:398–415, 2007.

[10] Y. Cheng and C.-W. Shu. Superconvergence and time evolution of discontinuous Galerkin finite element solutions. *J. Comput. Phys.*, 227:9612–9627, 2008.

[11] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc.*, 27:1–67, 1992.

[12] M. G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 277:1–42, 1983.

[13] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43:1–19, 1984.

[14] E. E.Carlini, F. R., and G. Russo. A weighted essentialy non oscillatory, large time-step scheme for Hamilton Jacobi equations. *SIAM J. Sci. Comp.*, 27(3):1071–1091, 2005.

[15] M. Falcone, T. Giorgi, and P. Loreti. Level sets of viscosity solutions : some applications to fronts and rendez-vous problems. *SIAM J. Appl. Math.*, 54(5):1335–1354, 1994.

[16] E. Godlewski and P.-A. Raviart. *Hyperbolic Systems of Conservation Laws*. SMAI. Ellipses, 1991.

[17] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comput.*, 67:73–85, 1998.

[18] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability preserving high order time discretization methods. *SIAM Rev.*, 43:89–112, 2001.

[19] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:666–690, 1999.

[20] G. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 1999.

[21] F. Lafon and S. Osher. High order two-dimensional nonoscillatory methods for solving Hamilton-Jacobi scalar equations. *J. Comput. Phys.*, 123:235–253, 1996.

[22] F. Li and C.-W. Shu. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations. *Appl. Math. Lett.*, 18:1204–1209, 2005.

[23] I. Mitchell. A toolbox of level set methods. *http://www.cs.ubc.ca/m̃itchell/ToolboxLS/*.

[24] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent Hamiliton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Automat. Control*, 50(7):947–957, 2005.

[25] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.
[26] S. Osher and C.-W. Shu. High order essentially non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 28:907–922, 1991.
[27] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
[28] Y.-T. Zhang and C.-W. Shu. High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.*, 24:1005–1030, 2003.

Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie 75252 Paris Cedex 05 France *and* UFR de Mathématiques, Site Chevaleret, Université Paris-Diderot, 75205 Paris Cedex France
E-mail address: boka@math.jussieu.fr

Department of Mathematics and ICES, University of Texas, Austin, TX 78712, USA
E-mail address: ycheng@math.utexas.edu

Division of Applied Mathematics, Brown University, Providence, RI 02912, USA
E-mail address: shu@dam.brown.edu